Tri_Group_inflation_comparison

February 12, 2025

1 Setup and Data

[1]:
```
from inflation_analysis import calculate_price_indexes, tri_grouping,␣
↪output_data, output_obs_table, price_index_over_time,␣
↪top_abs_weight_differences, top_price_index_contributors,␣
↪tri_grouping_extended
```

[2]:
```
# Parameters
start_year = 2015
end_year = 2022
data_folder="/Users/roykisluk/Downloads/Consumer_Expenditure_Survey/"
top_n = 10
base_year = start_year
comparison_year = end_year

# Grouping
demo, income, ses, total_mmb = tri_grouping_extended(start_year, end_year,␣
↪cex_data_folder = data_folder)
```

[3]:
```
# Prepare data: calculate price indexes for each group, secondary and primary␣
↪categories, and total
demo_analysis, demo_mmb = output_data(demo, start_year, end_year, base_year,␣
↪top_n, data_folder)
income_analysis, income_mmb = output_data(income, start_year, end_year,␣
↪base_year, top_n, data_folder)
ses_analysis, ses_mmb = output_data(ses, start_year, end_year, base_year,␣
↪top_n, data_folder)

# General population
print("Calculating price indexes for general population...")
gen_pop_df, gen_pop_secondary_df, gen_pop_primary_df,␣
↪gen_pop_yearly_price_index = calculate_price_indexes(start_year, end_year,␣
↪base_year, cex_data_folder=data_folder, verbose=False)
gen_pop = {
    'combined_secondary_df': gen_pop_secondary_df,
    'combined_primary_df': gen_pop_primary_df,
    'yearly_price_index': gen_pop_yearly_price_index
```

```
}
print("Done.")
```

```
Group 1/7 (Secular) started.

Loading price data: 100%|      | 8/8 [00:09<00:00,  1.13s/it]
Calculating price indexes: 100%|      | 8/8 [00:18<00:00,  2.31s/it]

Group 1/7 (Secular) successfully computed.
Group 2/7 (Conservative) started.

Loading price data: 100%|      | 8/8 [00:08<00:00,  1.05s/it]
Calculating price indexes: 100%|      | 8/8 [00:14<00:00,  1.82s/it]

Group 2/7 (Conservative) successfully computed.
Group 3/7 (Religious) started.

Loading price data: 100%|      | 8/8 [00:08<00:00,  1.08s/it]
Calculating price indexes: 100%|      | 8/8 [00:09<00:00,  1.18s/it]

Group 3/7 (Religious) successfully computed.
Group 4/7 (Haredi) started.

Loading price data: 100%|      | 8/8 [00:08<00:00,  1.06s/it]
Calculating price indexes: 100%|      | 8/8 [00:07<00:00,  1.09it/s]

Group 4/7 (Haredi) successfully computed.
Group 5/7 (Arabs) started.

Loading price data: 100%|      | 8/8 [00:08<00:00,  1.02s/it]
Calculating price indexes: 100%|      | 8/8 [00:08<00:00,  1.07s/it]

Group 5/7 (Arabs) successfully computed.
Group 6/7 (Young) started.

Loading price data: 100%|      | 8/8 [00:08<00:00,  1.05s/it]
Calculating price indexes: 100%|      | 8/8 [00:09<00:00,  1.20s/it]

Group 6/7 (Young) successfully computed.
Group 7/7 (Old) started.

Loading price data: 100%|      | 8/8 [00:08<00:00,  1.04s/it]
Calculating price indexes: 100%|      | 8/8 [00:10<00:00,  1.35s/it]

Group 7/7 (Old) successfully computed.
Group 1/10 (1) started.

Loading price data: 100%|      | 8/8 [00:07<00:00,  1.01it/s]
Calculating price indexes: 100%|      | 8/8 [00:06<00:00,  1.14it/s]

Group 1/10 (1) successfully computed.
Group 2/10 (2) started.

Loading price data: 100%|      | 8/8 [00:08<00:00,  1.02s/it]
Calculating price indexes: 100%|      | 8/8 [00:07<00:00,  1.07it/s]
```

```
Group 2/10 (2) successfully computed.
Group 3/10 (3) started.

Loading price data: 100%|      | 8/8 [00:08<00:00,  1.02s/it]
Calculating price indexes: 100%|      | 8/8 [00:08<00:00,  1.02s/it]

Group 3/10 (3) successfully computed.
Group 4/10 (4) started.

Loading price data: 100%|      | 8/8 [00:08<00:00,  1.03s/it]
Calculating price indexes: 100%|      | 8/8 [00:07<00:00,  1.05it/s]

Group 4/10 (4) successfully computed.
Group 5/10 (5) started.

Loading price data: 100%|      | 8/8 [00:08<00:00,  1.04s/it]
Calculating price indexes: 100%|      | 8/8 [00:07<00:00,  1.05it/s]

Group 5/10 (5) successfully computed.
Group 6/10 (6) started.

Loading price data: 100%|      | 8/8 [00:07<00:00,  1.00it/s]
Calculating price indexes: 100%|      | 8/8 [00:07<00:00,  1.07it/s]

Group 6/10 (6) successfully computed.
Group 7/10 (7) started.

Loading price data: 100%|      | 8/8 [00:08<00:00,  1.02s/it]
Calculating price indexes: 100%|      | 8/8 [00:07<00:00,  1.04it/s]

Group 7/10 (7) successfully computed.
Group 8/10 (8) started.

Loading price data: 100%|      | 8/8 [00:08<00:00,  1.01s/it]
Calculating price indexes: 100%|      | 8/8 [00:08<00:00,  1.04s/it]

Group 8/10 (8) successfully computed.
Group 9/10 (9) started.

Loading price data: 100%|      | 8/8 [00:09<00:00,  1.14s/it]
Calculating price indexes: 100%|      | 8/8 [00:08<00:00,  1.12s/it]

Group 9/10 (9) successfully computed.
Group 10/10 (10) started.

Loading price data: 100%|      | 8/8 [00:09<00:00,  1.13s/it]
Calculating price indexes: 100%|      | 8/8 [00:09<00:00,  1.15s/it]

Group 10/10 (10) successfully computed.
Group 1/5 (1) started.

Loading price data: 100%|      | 8/8 [00:09<00:00,  1.17s/it]
Calculating price indexes: 100%|      | 8/8 [00:07<00:00,  1.06it/s]

Group 1/5 (1) successfully computed.
Group 2/5 (2) started.
```

```
Loading price data: 100%|      | 8/8 [00:08<00:00,  1.02s/it]
Calculating price indexes: 100%|      | 8/8 [00:11<00:00,  1.41s/it]

Group 2/5 (2) successfully computed.
Group 3/5 (3) started.

Loading price data: 100%|      | 8/8 [00:08<00:00,  1.05s/it]
Calculating price indexes: 100%|      | 8/8 [00:16<00:00,  2.07s/it]

Group 3/5 (3) successfully computed.
Group 4/5 (4) started.

Loading price data: 100%|      | 8/8 [00:08<00:00,  1.08s/it]
Calculating price indexes: 100%|      | 8/8 [00:13<00:00,  1.71s/it]

Group 4/5 (4) successfully computed.
Group 5/5 (5) started.

Loading price data: 100%|      | 8/8 [00:08<00:00,  1.02s/it]
Calculating price indexes: 100%|      | 8/8 [00:04<00:00,  1.76it/s]

Group 5/5 (5) successfully computed.
Calculating price indexes for general population…

Loading price data: 100%|      | 8/8 [00:08<00:00,  1.01s/it]
Calculating price indexes: 100%|      | 8/8 [00:34<00:00,  4.36s/it]

Done.
```

2 Output

2.1 Tables

[4]:
```
# Observations tables
output_obs_table(start_year, end_year, demo_mmb)
```

```
+--------------+---------------+---------------+---------------+---------------
+---------------+---------------+---------------+---------------+
|              | 2015          | 2016          | 2017          | 2018          |
2019          | 2020          | 2021          | 2022          |
|--------------+---------------+---------------+---------------+---------------
+---------------+---------------+---------------+---------------|
| Secular      | 3736 (43.75%) | 4021 (45.24%) | 4114 (45.83%) | 4055 (46.34%) |
3668 (46.96%) | 2750 (49.71%) | 2690 (44.65%) | 2294 (41.99%) |
| Conservative | 2409 (28.21%) | 2395 (26.94%) | 2455 (27.35%) | 2370 (27.08%) |
2117 (27.1%)  | 1460 (26.39%) | 1577 (26.18%) | 1602 (29.32%) |
| Religious    | 1257 (14.72%) | 1244 (13.99%) | 1290 (14.37%) | 1185 (13.54%) |
1218 (15.59%) | 809 (14.62%)  | 1035 (17.18%) | 808 (14.79%)  |
| Haredi       | 734 (8.59%)   | 778 (8.75%)   | 757 (8.43%)   | 786 (8.98%)   |
565 (7.23%)   | 440 (7.95%)   | 551 (9.15%)   | 595 (10.89%)  |
| Arabs        | 1136 (13.3%)  | 1273 (14.32%) | 1327 (14.78%) | 1145 (13.08%) |
```

```
1103 (14.12%) | 513 (9.27%)   | 951 (15.79%)  | 727 (13.31%)  |
| Young        | 1340 (15.69%) | 1354 (15.23%) | 1358 (15.13%) | 1278 (14.6%)  |
1108 (14.19%) | 718 (12.98%)  | 877 (14.56%)  | 820 (15.01%)  |
| Old          | 2091 (24.48%) | 2372 (26.68%) | 2348 (26.16%) | 2375 (27.14%) |
2279 (29.18%) | 1786 (32.28%) | 1779 (29.53%) | 1663 (30.44%) |
| Total        | 8540 (100.0%) | 8889 (100.0%) | 8977 (100.0%) | 8751 (100.0%) |
7811 (100.0%) | 5532 (100.0%) | 6024 (100.0%) | 5463 (100.0%) |
+--------------+---------------+---------------+---------------+---------------
+---------------+---------------+---------------+---------------+
```

[5]:
```
output_obs_table(start_year, end_year, income_mmb)
```

```
+-------+---------------+---------------+---------------+---------------+-------
--------+---------------+---------------+---------------+
|       | 2015          | 2016          | 2017          | 2018          | 2019
| 2020          | 2021          | 2022          |
|-------+---------------+---------------+---------------+---------------+-------
--------+---------------+---------------+---------------|
| 1     | 873 (10.22%)  | 890 (10.01%)  | 940 (10.47%)  | 880 (10.06%)  | 646
(8.27%)   | 397 (7.18%)   | 503 (8.35%)   | 515 (9.43%)   |
| 2     | 929 (10.88%)  | 948 (10.66%)  | 1018 (11.34%) | 931 (10.64%)  | 710
(9.09%)   | 474 (8.57%)   | 517 (8.58%)   | 537 (9.83%)   |
| 3     | 927 (10.85%)  | 928 (10.44%)  | 916 (10.2%)   | 921 (10.52%)  | 708
(9.06%)   | 495 (8.95%)   | 548 (9.1%)    | 554 (10.14%)  |
| 4     | 862 (10.09%)  | 893 (10.05%)  | 911 (10.15%)  | 860 (9.83%)   | 759
(9.72%)   | 483 (8.73%)   | 563 (9.35%)   | 562 (10.29%)  |
| 5     | 832 (9.74%)   | 849 (9.55%)   | 874 (9.74%)   | 896 (10.24%)  | 748
(9.58%)   | 527 (9.53%)   | 589 (9.78%)   | 516 (9.45%)   |
| 6     | 816 (9.56%)   | 871 (9.8%)    | 869 (9.68%)   | 822 (9.39%)   | 776
(9.93%)   | 560 (10.12%)  | 608 (10.09%)  | 545 (9.98%)   |
| 7     | 830 (9.72%)   | 879 (9.89%)   | 863 (9.61%)   | 856 (9.78%)   | 781
(10.0%)   | 586 (10.59%)  | 595 (9.88%)   | 536 (9.81%)   |
| 8     | 814 (9.53%)   | 875 (9.84%)   | 885 (9.86%)   | 860 (9.83%)   | 851
(10.89%)  | 639 (11.55%)  | 655 (10.87%)  | 551 (10.09%)  |
| 9     | 818 (9.58%)   | 888 (9.99%)   | 861 (9.59%)   | 870 (9.94%)   | 891
(11.41%)  | 654 (11.82%)  | 712 (11.82%)  | 550 (10.07%)  |
| 10    | 849 (9.94%)   | 882 (9.92%)   | 880 (9.8%)    | 896 (10.24%)  | 957
(12.25%)  | 778 (14.06%)  | 767 (12.73%)  | 612 (11.2%)   |
| Total | 8540 (100.0%) | 8889 (100.0%) | 8977 (100.0%) | 8751 (100.0%) | 7811
(100.0%) | 5532 (100.0%) | 6024 (100.0%) | 5463 (100.0%) |
+-------+---------------+---------------+---------------+---------------+-------
--------+---------------+---------------+---------------+
```

[6]:
```
output_obs_table(start_year, end_year, ses_mmb)
```

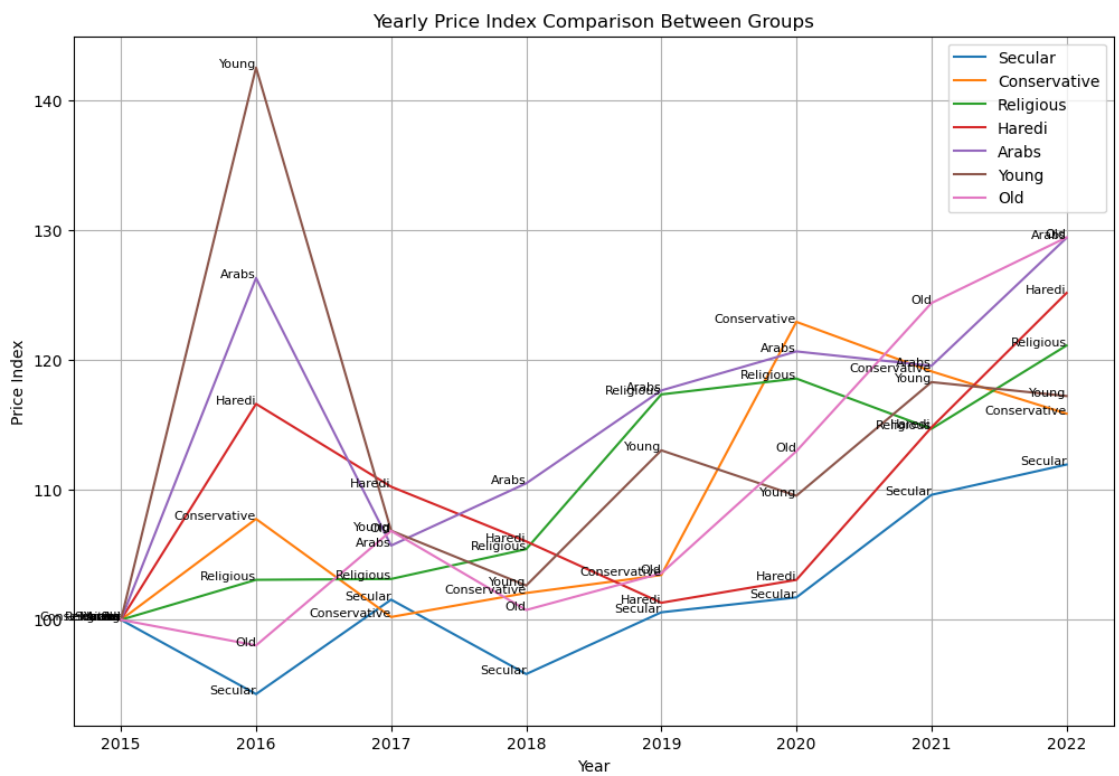
```
+-------+---------------+---------------+---------------+---------------+-------
--------+---------------+---------------+---------------+
|       | 2015          | 2016          | 2017          | 2018          | 2019
| 2020          | 2021          | 2022          |
```

```
|-------+---------------+---------------+---------------+---------------+-------
--------+---------------+---------------+---------------|
| 1     | 616 (7.21%)   | 695 (7.82%)   | 728 (8.11%)   | 654 (7.47%)   | 1146
(14.67%) | 807 (14.59%)  | 1204 (19.99%) | 1211 (22.17%) |
| 2     | 1916 (22.44%) | 2021 (22.74%) | 1977 (22.02%) | 1882 (21.51%) | 1017
(13.02%) | 564 (10.2%)   | 804 (13.35%)  | 753 (13.78%)  |
| 3     | 3495 (40.93%) | 3523 (39.63%) | 3627 (40.4%)  | 3624 (41.41%) | 1911
(24.47%) | 1490 (26.93%) | 1503 (24.95%) | 1352 (24.75%) |
| 4     | 2402 (28.13%) | 2569 (28.9%)  | 2583 (28.77%) | 2505 (28.63%) | 3564
(45.63%) | 2556 (46.2%)  | 2393 (39.72%) | 2016 (36.9%)  |
| 5     | 121 (1.42%)   | 95 (1.07%)    | 102 (1.14%)   | 127 (1.45%)   | 189
(2.42%)   | 163 (2.95%)   | 153 (2.54%)   | 146 (2.67%)   |
| Total | 8540 (100.0%) | 8889 (100.0%) | 8977 (100.0%) | 8751 (100.0%) | 7811
(100.0%) | 5532 (100.0%) | 6024 (100.0%) | 5463 (100.0%) |
+-------+---------------+---------------+---------------+---------------+-------
--------+---------------+---------------+---------------+
```

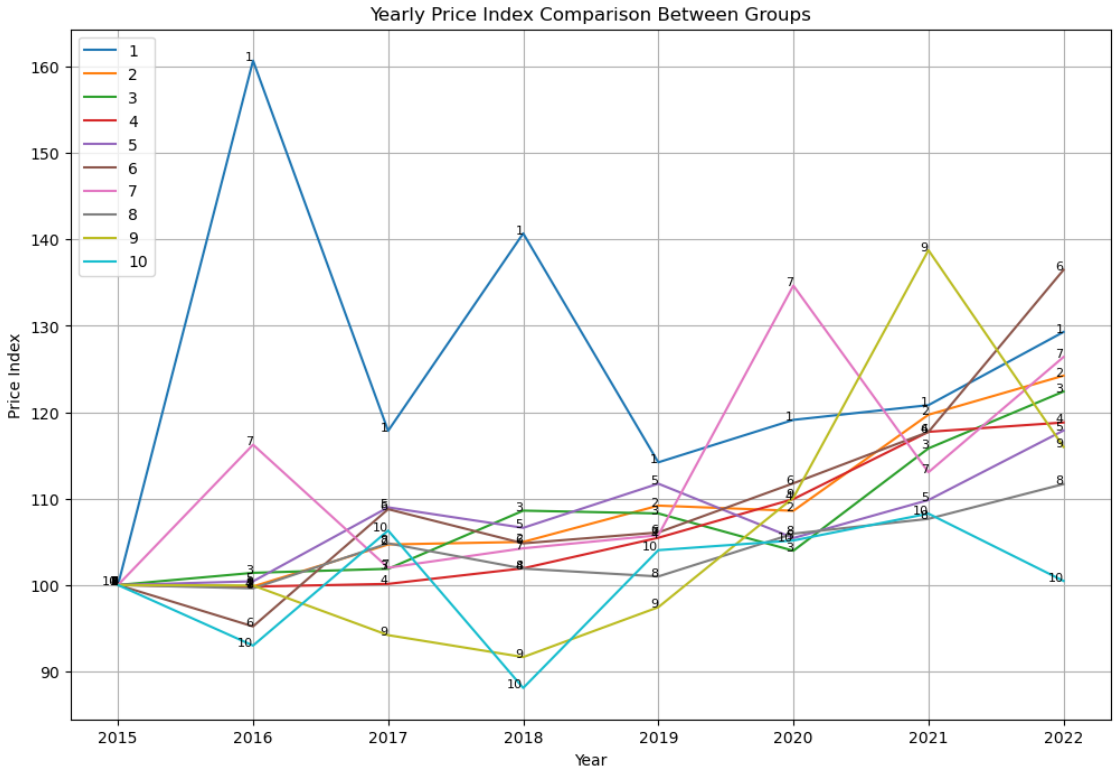
2.2 Plots

2.2.1 Yearly Price Index Comparison Between Groups

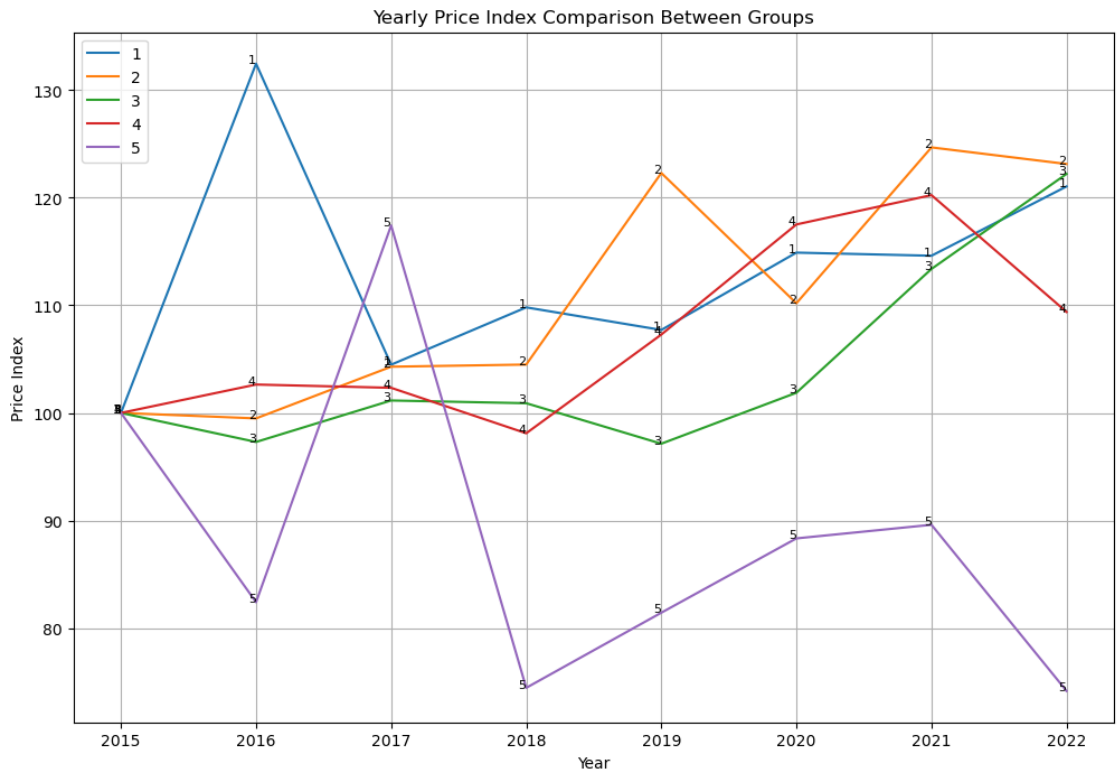
[7]:
```
price_index_over_time(demo_analysis)
```



[8]:
```
price_index_over_time(income_analysis)
```



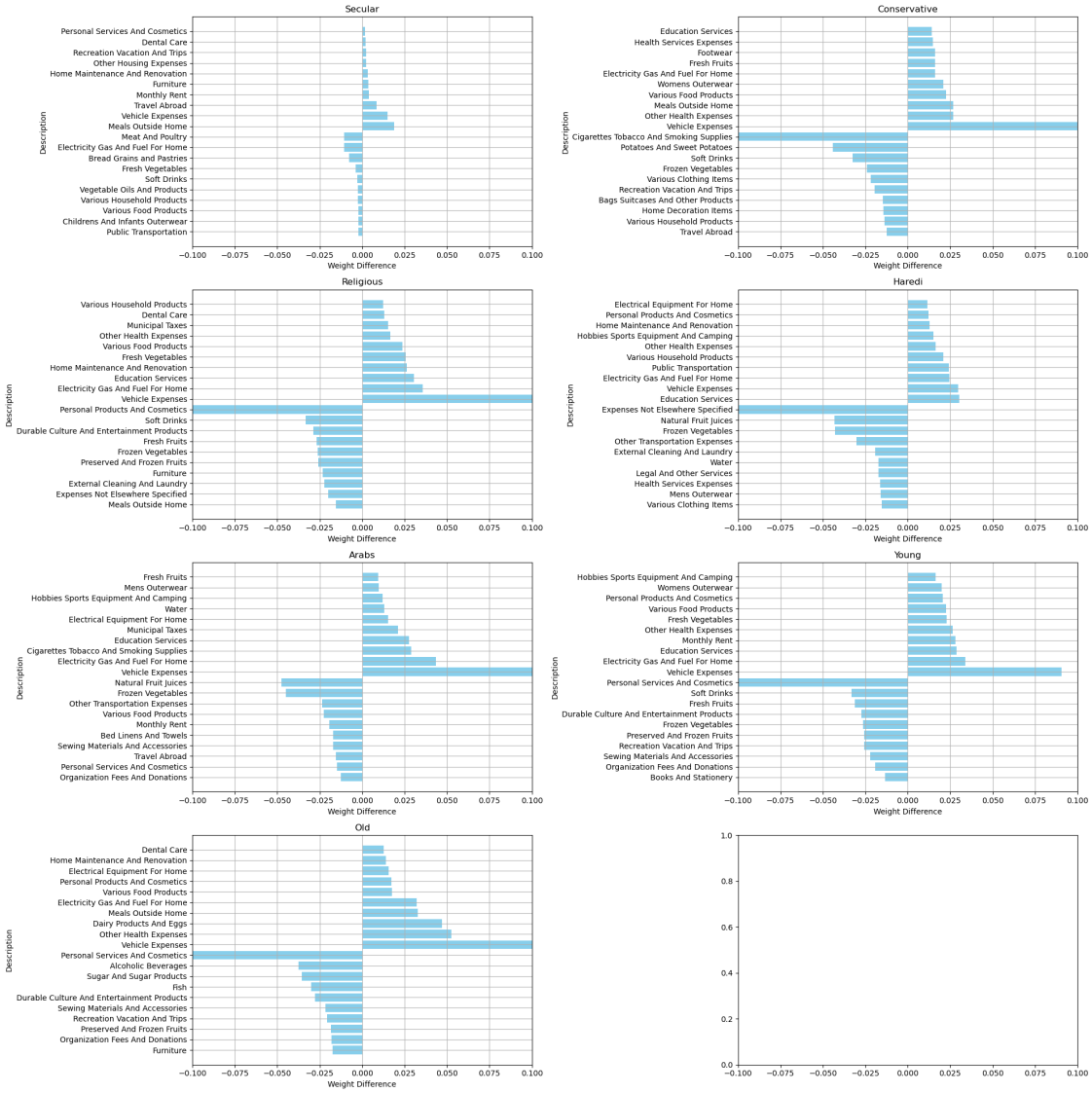
[9]:
```
price_index_over_time(ses_analysis)
```



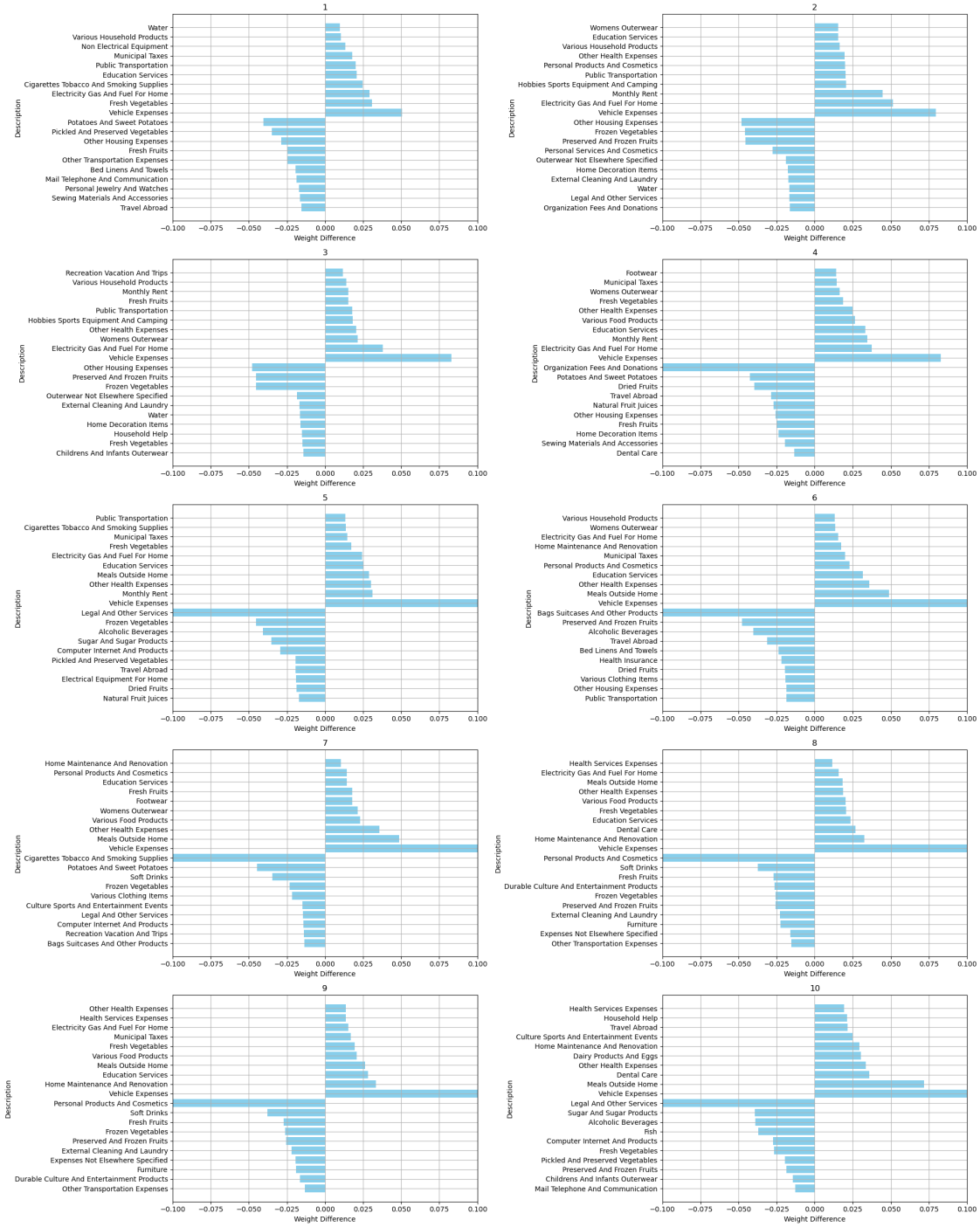
2.2.2 Top Weight Differences

[10]:
```
# Define control group
weights_comparison_control =␣
↪gen_pop['combined_secondary_df'][gen_pop['combined_secondary_df']['Year'] ==␣
↪comparison_year]
```

[11]:
```
# Top weight differences - demograhic groups
demo_comparison_groups = {}
for group in demo_analysis:
    demo_comparison_groups[group] =␣
↪demo_analysis[group]['combined_secondary_df'][demo_analysis[group]['combined_secondary_df']
↪== comparison_year]
top_abs_weight_differences(demo_comparison_groups, weights_comparison_control,␣
↪top_n)
```

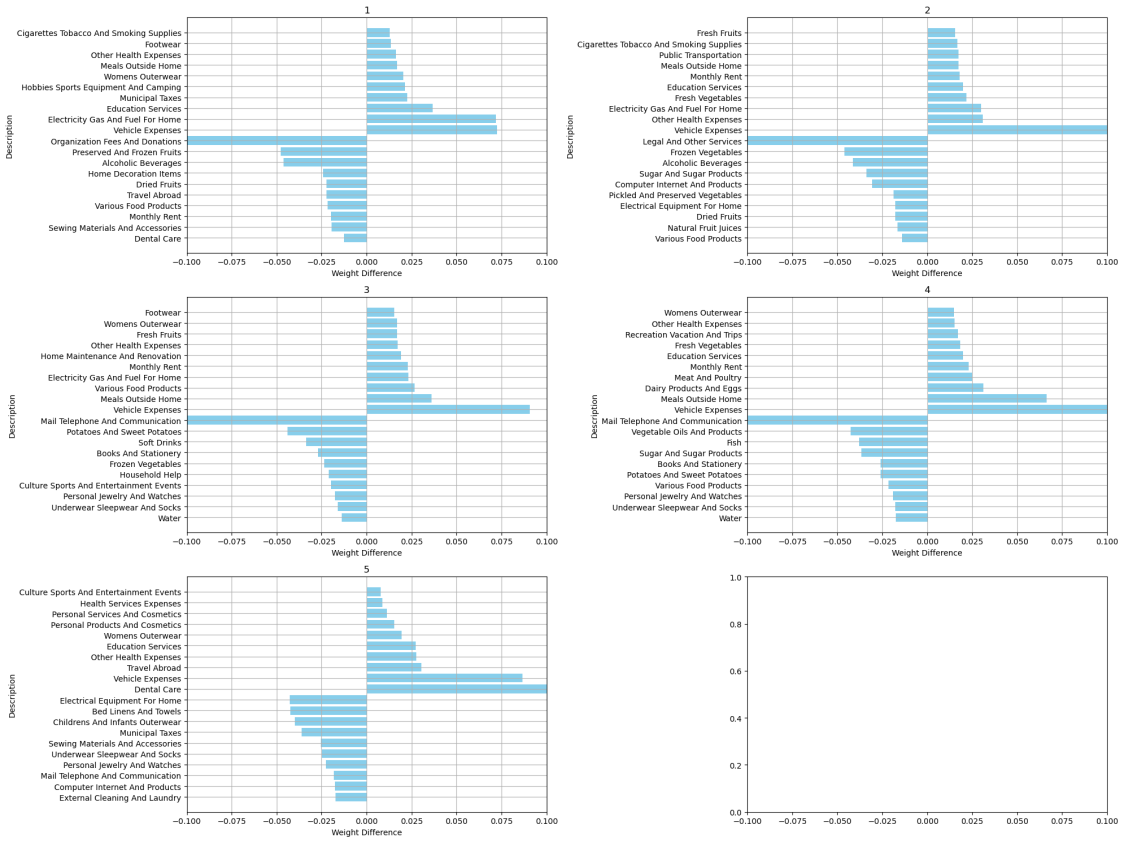


[12]:
```
# Top weight differences - income groups
income_comparison_groups = {}
for group in income_analysis:
    income_comparison_groups[group] =␣
 ↪income_analysis[group]['combined_secondary_df'][income_analysis[group]['combined_secondary_
 ↪== comparison_year]
top_abs_weight_differences(income_comparison_groups,␣
 ↪weights_comparison_control, top_n)
```



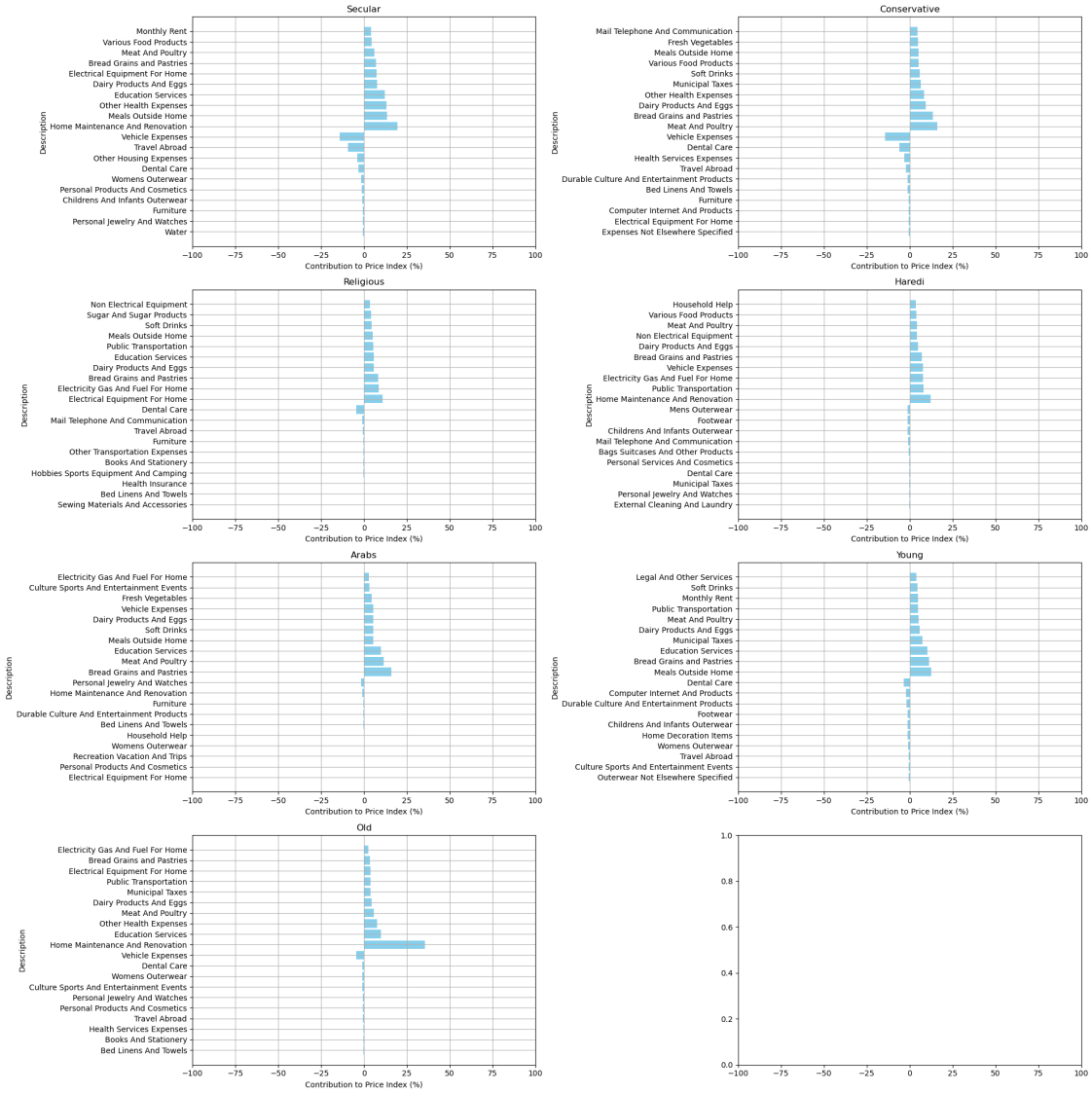
[13]:
```
# Top weight differences - SES groups
ses_comparison_groups = {}
for group in ses_analysis:
```

```
    ses_comparison_groups[group] =␣
↪ses_analysis[group]['combined_secondary_df'][ses_analysis[group]['combined_secondary_df']['
↪== comparison_year]
top_abs_weight_differences(ses_comparison_groups, weights_comparison_control,␣
↪top_n)
```

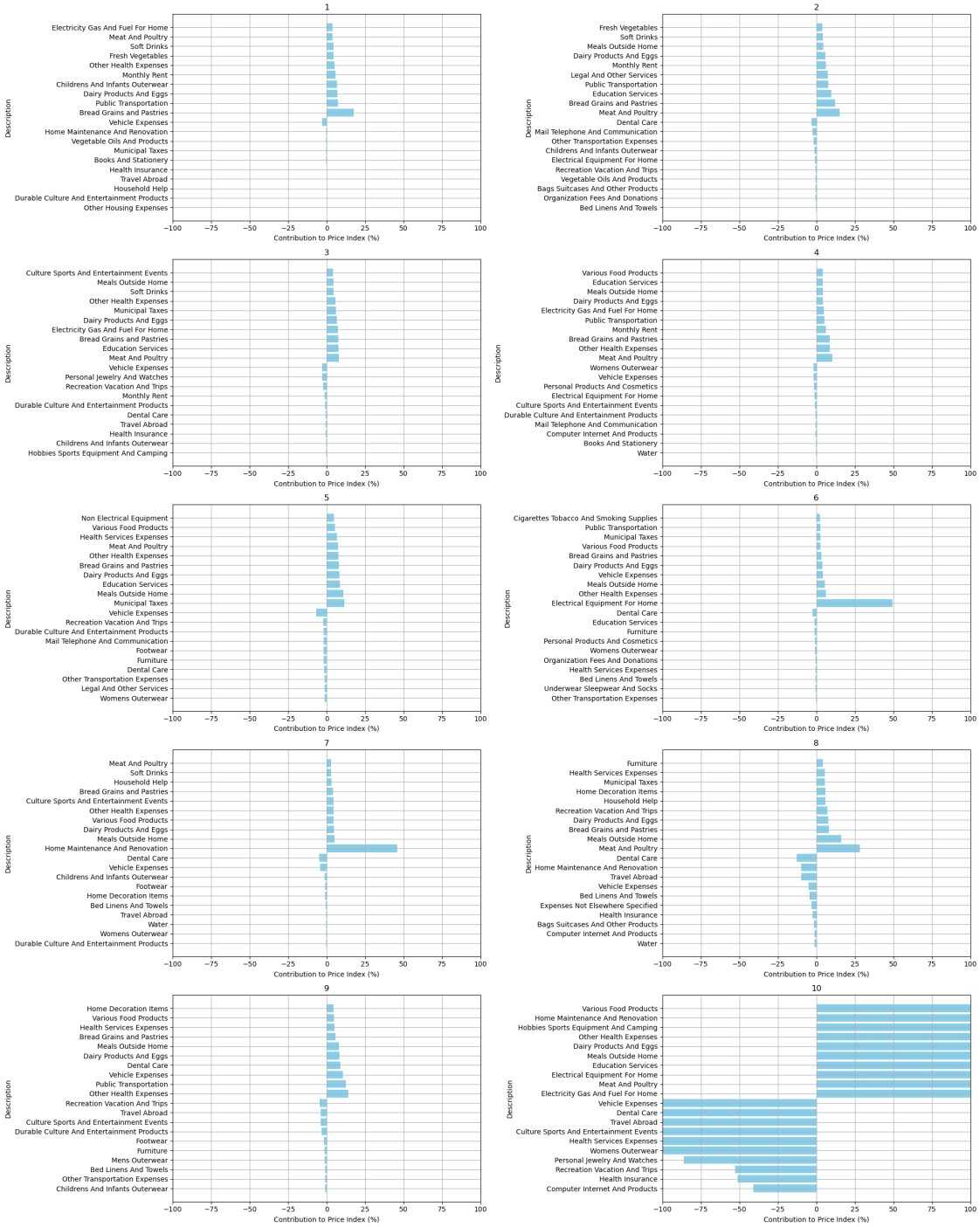


2.2.3 Top Contributors to CPI Change

[14]:
```
# Top contributors - demographic groups
demo_yearly_price_indexes = {}
for group in demo_analysis:
    demo_yearly_price_indexes[group] =␣
↪demo_analysis[group]['yearly_price_index'][comparison_year]
top_price_index_contributors(demo_comparison_groups, demo_yearly_price_indexes,␣
↪top_n)
```



[15]:
```
# Top contributors - income groups
income_yearly_price_indexes = {}
for group in income_analysis:
    income_yearly_price_indexes[group] =␣
 ↪income_analysis[group]['yearly_price_index'][comparison_year]
top_price_index_contributors(income_comparison_groups,␣
 ↪income_yearly_price_indexes, top_n)
```



[16]:
```
# Top contributors - SES groups
ses_yearly_price_indexes = {}
for group in ses_analysis:
    ses_yearly_price_indexes[group] =␣
↪ses_analysis[group]['yearly_price_index'][comparison_year]
```

```
top_price_index_contributors(ses_comparison_groups, ses_yearly_price_indexes,␣
↪top_n)
```